

The purge module clears content from reverse proxy caches like [Varnish](#), [Squid](#) or [Nginx](#) by issuing an HTTP PURGE request to them. This allows delivering content updates faster to end users while caching efficiently.

The current stable 1.x branches work in conjunction with the [cache expiration module](#) to act upon events that are likely to expire URLs from the proxy cache. It will not longer receive new features, just bugfixes.

Requirements

- One or more [reverse proxy caches](#) like [Varnish](#) (recommended), [Squid](#) or [this section in the Varnish chapter](#) that point to your webserver(s). Varnish needs a modification to its configuration file. See [this section in the Varnish chapter](#) of the Drupal handbook.
- Squid needs to have purging [enabled in its configuration](#).
- Nginx needs [an extra module and configuration](#). See the installation hints below and the included the README, also see this [issue #1048000](#) for more background info and compiling/installation hints.
- A cachable version for Drupal 6. This can be an official Drupal 6 release with [a patch](#) applied or use [Pressflow](#), a cachable friendly fork of Drupal. Drupal 7 works out of the box.
- PHP with [curl](#) enabled. The 1.x releases of Purge uses curl for issuing the http PURGE requests.

Installation

- Unpack, place and enable just like any other module.
- Navigate to Administration -> Site configuration -> Purge settings.
- Set your proxy URL(s) like `http://localhost` or `http://192.168.1.23:8080` `http://192.168.2.34`.
- If you are using nginx you need to specify the purge path and the get method in your proxy setting like this: `http://192.168.1.76:8080/purge?purge_method=get`
- If you are using the Acquia Cloud we recommend you use the platform specific module [Acquia Purge](#) instead.
- Optional: Install [Rules](#) for advanced cache clearing scenarios or [Drush](#) for command line purging. Both are supported through the expire module.

Q&A

How do I know if its working?

Purge reports errors to watchdog. Also when running "varnishlog" on the proxy your should see PURGE requests scrolling by when you (for instance) update an existing node.

How can I test this more efficiently?

The expire module has drush support so you can issue purge commands from the command line. See [#1054584](#) You can also test if your proxy is configured correctly by issuing a curl command in a shell on any machine in the access list of your proxy: `curl -X PURGE -H "Host: example.com" http://192.168.1.23/node/2`

Why choose this over the [Varnish module](#)?

Purge just issues purge requests to your proxy server(s) over standard http on every url the expire module detects. It requires modification of your Varnish configuration file.

The varnish module has more internal logic to purge your Varnish cache completely, which can be more

disruptive than the expire module integration it also offers. It uses a terminal interface to communicate to Varnish instead of HTTP. This allows for more features but also hands over full control over the Varnish instance to any machine having network access to it. (This is a limitation of Varnish.) Also firewall or other security policies could pose a problem. It does not require modification of your config file. If you have the choice Varnish module is probably your best bet but Purge might help you out in places where Varnish module is not an option.

What happened to support for Acquia Cloud platforms?

Although this was supported in earlier versions of the 1.x series, we've dropped this in favor of the [Acquia Purge](#) module which gives a smoother and more integrated experience.